

Team 14 Black hole

112550031 陳炤宇

112550039 李宗諺

112550139 簡士原

1. Introduction

本作業以黑洞為主題，目標是在一個可互動視角的黑洞模擬場景中加入具敘事性的動畫元素。我們以既有的黑洞背景渲染為基礎，將紅色光點設定為繞黑洞公轉的主要角色，並以幾何著色器把沿途取樣的軌跡點擴增為發光粒子群，形成清楚可辨識的環繞軌道與能量尾跡。同時，我們加入由使用者觸發的彗星事件：按鍵即可從目前攝影機方向發射一束白色彗星並朝黑洞飛行，讓畫面從單一的穩態運動延伸為「持續公轉的系統」與「偶發事件」並存的動態場景。整體設計強調以 shader 為核心完成動畫視覺效果，並在有限頁數與時間內以清楚的資料流與可重現的操作方式呈現成果。

背景故事：

黑洞周圍的可見發光可概略分為兩種來源。第一種是短時間內就會改變或消失的暫態亮點：畫面中的紅色光團可視為低密度的等離子體或塵埃熱點，會在潮汐撕裂與吸積流的作用下迅速被拉散並混入周圍物質，因此亮度在短時間內明顯衰減；固定的黃色光團則對應較遠處的背景輻射，其可見性取決於光線穿越吸積盤的路徑，當視線落在高密度區域時，可能因吸收或散射而呈現短暫遮蔽。第二種則是相對穩定、可長時間維持的結構性發光：公轉的黃色光線代表由磁場導引的高能束流（或場線的視覺化），因為能量可持續注入，軌跡能在較長時間內保持清晰，並隨著發射源繞黑洞運動而同步公轉。至於按下 M 觸發的白色彗星，則被設定為一個已經進入捕獲區的外來天體碎片：它從觀察者視角切入場景，沿重力井快速向內墜落並在接近黑洞時逐步拉長成明顯的尾跡，用來補充系統中「偶發入侵」的事件型動態，並與持續公轉的背景結構形成對比。

2. Implementation Details

1. Render Pipeline Overview

本專題以黑洞模擬為基礎：使用 Compute Shader 產生影像，再以全螢幕 Quad 顯示背景。為符合 HW4 「Animation + Geometry Shader」主題，我們加入兩個動態效果：

Orbit Trail：紅色物件繞黑洞公轉並留下發光粒子軌跡。

Meteor：按鍵觸發從攝影機方向發射、朝黑洞飛行的彗星與尾跡。

每幀流程簡述：

- 更新物件位置（公轉 / 彗星運動）
- 將最新位置寫入 trail buffer，更新到 GPU VBO
- 繪製黑洞背景、格線
- 以 Geometry Shader 繪製 orbit trail 與 meteor trail（發光粒子）

2. Trail Data Structure & GPU Update (C++)

Trail 使用固定長度的取樣點（例如 512 點），CPU 端以 `vec4(pos, spawnTime)` 存放：

- xyz：世界座標取樣點
- w：該點寫入時刻（秒）

更新策略採「最新點在前」的序列（其餘往後推），接著以 `glBufferSubData` 將整段資料更新進 VBO（dynamic update）。此設計讓 shader 能同時利用：

- 序列位置（新→舊）做尾端漸淡
- 實際時間差（`now - spawnTime`）做存活衰減

3. Geometry Shader: Points → Polygons

Orbit trail 以 `GL_POINTS` 作為輸入；每個點在 Geometry Shader 中被擴增為 polygon（billboard quad），透過 `triangle_strip + EmitVertex/EndPrimitive` 生成面片，符合作業要求「使用 Geometry Shader 產生新的 point/line/polygon」。

- 為使效果更明顯、非單一線條，我們在 GS 中對同一個軌跡取樣點再生成多顆「衛星粒子」：

每個取樣點周圍生成 `SAT_COUNT (≥ 5)` 個小粒子 quad

- 粒子中心以半徑 R 圍繞取樣點形成環狀分布，並加入少量 jitter 避免重疊
- 以 `uTime` 讓這些小粒子在取樣點周圍持續自轉，形成「主軌道繞黑洞公轉 + 局部繞紅光旋轉」的雙層運動視覺

Trail 漸層與衰減（在 GS 中計算）：

- 以 $gl_PrimitiveIDIn / (uPointCount-1)$ 得到沿 trail 的新舊比例（新端亮、尾端淡）
- 以 $\exp(-k * (uTime - spawnTime))$ 控制時間消散

4. Fragment Shader: Soft Glow Particle

Fragment Shader 以 quad 的 UV 計算類 Gaussian mask (中心亮、邊緣柔和衰減)，並乘上 GS 傳入的 alpha。對極小 alpha 直接 discard 以降低 fill-rate。顏色由 head/tail 顏色混合形成漸層，使頭端偏亮、尾端偏暗。

5. Blending / Depth Settings (可視性強化)

為呈現「光」的堆疊效果，trail 繪製時使用 additive blending (例如 SRC_ALPHA, ONE)，並在粒子繪製階段調整 depth test/ordering 以避免被背景覆蓋；繪製結束後恢復一般狀態，避免影響其他 pass。

6. Meteor (Key-triggered effect) & Separation from Orbit Trail

更按鍵觸發的彗星由攝影機附近生成，速度方向朝黑洞並可分 fast/slow 模式。Meteor 的尾跡使用獨立的 VAO/VBO 與 trail buffer 更新流程，使其可設定不同粒子大小與 head/tail 顏色。

若 orbit trail 與 meteor 共用同一組 GS，會出現「彗星也被展開成一圈多顆粒子」的耦合問題；因此可用 uniform 模式切換 (例如 uMode)：

- uMode=1 : orbit trail 多顆展開模式
- uMode=0 : meteor 單顆/簡化模式

讓同一套 shader program 在不同 draw call 下呈現不同視覺語意。

3. Discussion

Proposer(陳炤宇)想要從原專案的黑洞基礎上加入更真實的物理模擬效果，使用 geometry shader 產生從四周湧向黑洞旋轉吸入的小行星們與描繪行經的軌跡，過程中會依照現實物理原理，產生光線與小行星一同被吸入黑洞的視覺效果，在小行星撞擊時也產生湮滅的效果。

Critic(李宗諺)提出影響 project 進行的重要問題，包括複雜且龐大的粒子光追效果交互作用下，會導致系統運算效能崩潰，以及視窗尺寸與時間限制導致畫面沒有重點且雜亂。

Negotiator(簡士原)從雙方的意見中取出折衷的方案，讓小行星的生成變為人為控制，並將繞行的成分交給持續在外頭的紅色星球，用更少的資源與控制表現行星與黑洞的相對關係，讓視覺效果以實務上可行的方式呈現。

4. Uniqueness / Creativity

本專題以開源專案 **black_hole** 作為基底，保留其黑洞重力透鏡與吸積盤等核心視覺，並在不破壞原場景風格的前提下，加入「可被觀察、可被觸發、可形成事件對比」的動畫層，使作品從單純的物理視覺展示轉為具有敘事節奏的動態場景。我們將畫面中的光源分工成不同角色：紅色光團作為持續公轉的主體，軌跡則以發光粒子群呈現其能量路徑；而按鍵觸發的白色彗星則作為「外來且已進入捕獲區」的短暫事件，讓觀察者能在同一套黑洞環境中引入一次性的擾動，形成「穩態系統」與「偶發入侵」的對照。這種把互動事件嵌入黑洞視覺主體的做法，是相較原始專案更偏向動畫作業目標的再詮釋。

在實作策略上，我們選擇 **Scheme A (Geometry Shader)** 作為軌跡視覺的主要表現手段，將軌跡視為一串時序取樣點，再以 shader 在 GPU 端轉換為具體可見的發光粒子效果。與常見「CPU 端直接生大量粒子並逐顆更新」的作法不同，我們將「視覺複雜度」集中在 shader 的生成規則與參數化控制上，讓同一套效果可透過少量 uniform (例如顏色、大小、模式) 切換成不同語意：一般公轉軌跡與彗星事件在視覺上可明確區分，但仍維持整體風格一致。整體而言，我們的獨特性不在於替換原有黑洞渲染，而在於以最小侵入方式，把「軌跡、粒子、事件觸發」整合為符合 HW4 主題的動畫設計，並以清楚的角色分工與互動操作提升作品的可讀性與可玩性。

5. Work Assignment

112550031 陳炤宇：軌道與尾跡效果、部分 report 撰寫、管理 GitHub

112550039 李宗諺：撰寫微粒軌道與尾跡、調整最後畫面呈現參數、撰寫 report 內 Introduction、Implementation Details、Uniqueness / Creativity 的初稿

112550139 簡士原：Meteor 設計、部分 report 撰寫、README 撰寫

6. References

原作品引用自 https://github.com/kavan010/black_hole

7. Results

Demo 影片連結：<https://youtu.be/3DbLwpLOKDA>

GitHub repo: https://github.com/cychobby/ICG_HW4_Group14.git